(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 April 2002 (11.04.2002)

PCT

(10) International Publication Number
WO 02/29715 A1

(51) International Patent Classification⁷: G06N 3/00, G06F 9/44

(21) International Application Number: PCT/SG00/00166

(22) International Filing Date: 3 October 2000 (03.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (*for all designated States except US*): **KENT RIDGE DIGITAL LABS** [SG/SG]; 21 Heng Mui Keng Terrace, Singapore 119613 (SG).

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **SITIRAM, Ranganatha** [IN/SG]; Block 218, #02-250 Choa Chu Kang Central, Singapore 680218 (SG). **ANNAPOORNA, Nayak, Pangal** [IN/SG]; Block 234, #07-05 Choa Chu Kang Central, Singapore 680234 (SG).

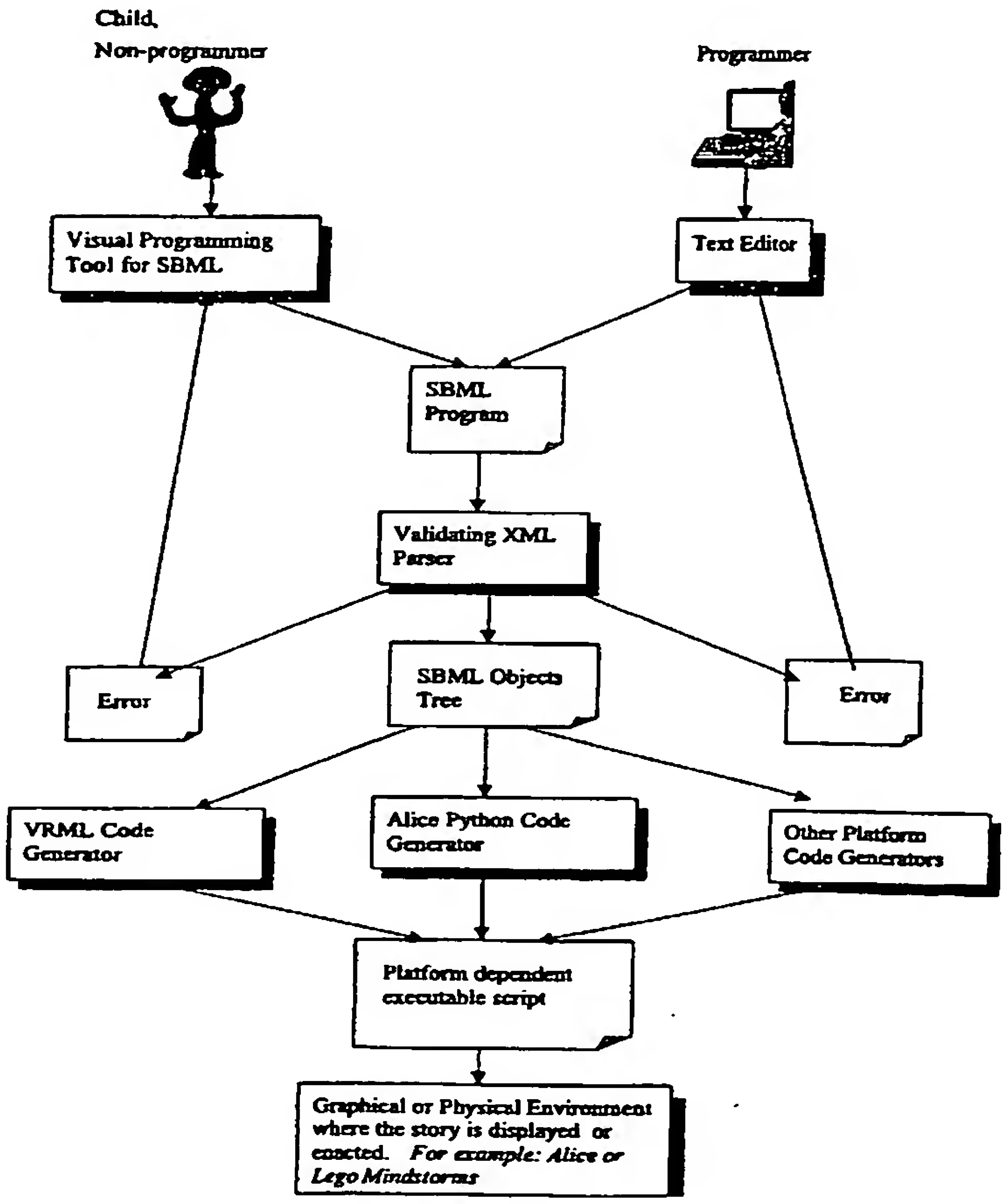
(74) Agents: **JACOB, Sheena, R.** et al.; Alban Tay Mahtani & De Silva, Raffles City Post Office, P.O. Box 0643, Singapore 911722 (SG).

(81) Designated States (*national*): CN, GB, IN, JP, SG, US.

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A SYSTEM, METHOD AND LANGUAGE FOR PROGRAMMING BEHAVIOUR IN SYNTHETIC CREATURES


Child,
Non-programmer
Programmer
Visual Programming Tool for SBML
Text Editor
SBML Program
Validating XML Parser
Error
SBML Objects Tree
Error
VRML Code Generator
Alice Python Code Generator
Other Platform Code Generators
Platform dependent executable script
Graphical or Physical Environment where the story is displayed or enacted. *For example: Alice or Lego Mindstorms*

(57) Abstract: The invention consists of a mark-up language to program behaviour of *Synthetic Creatures*. It can also be used for creating digital stories in graphical environments, virtual worlds, interactive toy systems, and robots. It is furhter designed specify spatial and temporal unfolding of life-like behaviour in synthetic creatures, while imitating the behaviours, actions and expressions of humans and other living creatures.

WO 02/29715 A1

A SYSTEM, METHOD AND LANGUAGE FOR PROGRAMMING BEHAVIOUR IN SYNTHETIC CREATURES

Field of the invention

The present invention relates to a system, method and new mark-up language to program behaviour of, for example, graphical characters, interactive toys and robots, and may, if desired, be used for creating digital stories in graphical environments, virtual worlds, interactive toy systems, and robots.

Definitions

Throughout this specification a reference to a synthetic creature is to be taken as including a graphical character, digital character, toy, interactive toy, robot, industrial robot; and all or part of a real character (human or animal) captured in digital image, on film, video tape, or otherwise.

Background to the invention

The field of interactive toys and entertainment robotics has been advancing by leaps and bounds. Toys are available now that are embedded with computing elements, sensors and actuators, that can interact with the surroundings and display interesting behaviour. "Furby" (trademark) by Tiger Electronics is a good example. However, these toys are limited due to their lack of programmability. "Furby" can not be reprogrammed to display what the owner likes or wants.

Robotic kits are available that have a programming language of their own. An example is the "Lego" (trademark) "Mindstorms" (trademark) robotic kit meant for children to learn design, construction and programming in an entertaining manner. However, the RCX programming language that "Mindstorms" uses is a general purpose language, and has a relatively low-level to enable the programming of behaviour in the robot. For example, with RCX a child can instruct a motor to operate for a certain period of time so that the robot can move accordingly. To achieve more complex behaviour such as, for example, walking of a dog-like robot, requires a lot more knowledge, skill and time for the programmer.

It is desirable to have a high-level robotic language to program behaviour that one sees in the real world such as for example, walking, waving, winking, and so forth. With such a language, more complex and interesting behaviour can be easily programmed for the amusement of children, such as singing and dancing, acting a scene, conversation amongst a group of toys, a game, mock combat, and so forth.

Further, many robot languages are tied to the specific hardware of a particular robot. As a result a program written to control the behaviour of one type of robot cannot be used in other robots built on different hardware platforms. There is a need for a high-level language to program behaviour in toys and robots.

With rapid advances in semiconductor technology, micro-sensors and actuators and system-on-chip technology, more sophisticated toy mechanisms will be developed that were hard to achieve in industrial robots a decade ago.

Consideration of prior art

US Patent 5805784

The disclosure of this patent is directed at the storyline of a dynamically generated entertainment program, such as a video game, which is generated using a matrix of reusable storyline fragments called substories. A set of characters that participate in the storyline established and a set of reusable substories is defined. Each substory represents a “fragment of a story”, usually involving an action by a subject, where the subject is one of the characters. Most substories can be reused multiple times with different ones of the characters being the subject and different ones of the characters being the direct object of the substory. Each substory has a set of possible reaction substories, which are a subset of the defined substories. A plan list stores plan data indicating ones of the substories to be performed at specified times. An initial “seed story” in the form of an initial set of substories is stored in the plan list. The substories stored in the plan list are executed at times corresponding to their respective specified times. For at least a subset of the executed substories, the end user of the system is either shown a video image representing the executed substory or is otherwise informed of the executed substory. In reaction to each executed substory, plans to perform additional substories are generated. The additional substories are from a set of possible reaction substories for each executed story. Each plan to perform an additional substory is assigned to a specified time, and plan data representing the plan is stored in the plan list.

The substories could be video or image sequences. The specification discloses how substories are related. However, this is related to stories on a multimedia screen, and is not applicable to toys and robots. This specification also does not disclose a high-level markup language for specifying a story.

WO9303453

This discloses a system for the sequential performance of a prerecorded story including text, animations or video, and audio information. The system, preferably implemented in a personal computer, has a continuous mode, in which it performs the story linearly and unbroken; a wait mode, in which it performs loops of animation, or otherwise is on standby for commands from the user; and an interactive mode, in which the system performs animations, sound or other activities which are tangential to the linear story. Text is displayed on a screen of the system, along with graphics and/or video. The text is pronounced by the system in the course of the sequential performance. When the computer is in the interactive mode the user may command it to repeat words which are in the text. The pronunciation of the words is the same as the pronunciation in the originally pronounced context. In both the continuous mode and the interactive mode, the pronounced words are highlighted. Certain animations are made inaccessible to the user, even in the interactive mode, until the user has executed certain prerequisite steps. Therefore, certain animations are interdependent or nested. The performance of a given animation may depend on whether a particular action has been carried out, or on whether another animation has already been performed, or on a random factor generated by the computer.

US5966526

This patent has the stated purpose of providing a simulation device for fostering a virtual creature. A virtual creature is grown while disciplining or training of the virtual creature occurs when a player conducts a corresponding treatment in response to a call or a request from the virtual creature on the screen. The device is provided with mark display units displaying a plurality of kinds of treatments in respect of fostering the virtual creature. This is by marks and key switches for inputting a corresponding treatment by selecting a specific mark from the plurality of marks. The device is provided with a storing unit for storing control data in respect of fostering the virtual creature, a control unit reading corresponding control data from the storing unit when the treatment in respect of fostering the virtual creature is inputted by operating the key switches. The device conduct control processing in respect of fostering the virtual creature based on the read control data, and has with a display unit for displaying the fostered virtual creature.

The objectives of the present invention and this patent are quite different. The present invention provides a programming means for specifying behaviour of virtual creatures in a high-level markup language. This patent specification provides a simulation device for fostering/growing virtual creatures.

WO9316776

A real-time, interactive, motion-based, simulator entertainment system that employs a computer generated video game (or network of video games) that interacts with a motion-based, operator-controlled control station or simulator. The system employs a computer processor, helmet-mounted projection display, a motion-based cockpit, control yokes or joy sticks, a sound system, and computer-generated video games. A plurality of participants (typically two) interact with selective and timed video scenarios to achieve an objective. Interaction is achieved using the control yokes and buttons. Each simulator operates independently of the other, except that group of participants may play the same scenario, possibly at the same time by ganging or networking sets of simulators. Each motion-based simulator is designed and cosmetically enhanced to appear as an armed space vehicle, for example, and comprises an interactive, video scenario virtually displayed on a windshield screen allowing the participants to interact with the system to achieve the predetermined game objective. The simulator system incorporates selection devices (yokes or joy-sticks), display networks, and selection buttons and controls that permit interaction with the system by the participants in response to information from the scenarios presented on the display.

This is a simulator environment for video/image games, and has very little in common with the present invention in terms of behaviour programming.

US5768122

An object oriented approach to virtual motion programming utilizes generic motion identifiers such as "turn" and "accelerate" in combination with modifiers such as the start and end times, the magnitude, and the direction of the motion object to be simulated to construct universal generic motion descriptions of complex motion events. The minimal bandwidth requirements of this virtual motion programming approach enables use of MIDI communication standards for virtual reality show systems and interactive computer applications, including video games. A virtual motion controller receives generic motion descriptions and breaks down the descriptions into their constituent parts. The controller expands the generic motion identifiers into general acceleration profiles and processes the general acceleration profiles with the modifiers to generate ideal acceleration profiles that define the acceleration components for the generic motion description with reference to an ideal motion platform having unlimited mechanical movement. The controller is configured with artificial intelligence for learning the capabilities of a particular motion platform. The controller translates the ideal acceleration profiles into customized position commands for the motion platform based upon the controller's knowledge of the platform's capabilities. The controller is configured with an offset processor for optimizing the position of the platform for maximum simulation of an ideal acceleration profile in the future. Position processors configured with fuzzy logic convert the position commands from the controller into control signals for the actuator motors of the platform, and prevents the actuators from overextending.

This patent discloses a high-level language/specification of generic descriptors for motion programming/control of robots.

US5617515

A hand-held controller wand including three angular rate measurement sensors is coupled to a computational device for translating roll, pitch, and yaw data into translation and rotation signals or commands that effect the movement of an external device's control point, which may be an end effector of a robot. A transformation matrix is continually updated from the angular rate information obtained from the hand-held controller. This matrix is used to update a direction vector. A value may be stored in the computation device corresponding to a predetermined speed of movement. A button or other convenient control allows the operator to control translational movement of the end effector or other controlled device in a direction defined by the orientation of the hand held controller at the predetermined speed, thus providing an intuitive control input for real-time direction and programming of movement. Because the present orientation of the wand is also known to the computational device, rotational movement may also be directed. Translation and orientation motion input may be provided simultaneously or separately, in accordance with the programming and/or under control of the computational device. The computational device translates the computed direction vector into signals corresponding to commands that depend upon the device being controlled; such translation is especially suited for robot controllers, because such controllers accept simple movement commands that may readily be computed from the direction vectors. However, the controller is also useful for other applications in which motion information is required, including the motion of real objects and virtual reality.

Again, this is limited to low level motion control of robots. It does not cover high-level behaviours such as, for example, facial expression, gestures, and so forth.

US5392207

A programming aid for troubleshooting real-time motion control programs controlling servo-motors employs a graphical control language in which functionally related groups of instructions are represented by icons displayed on a programming terminal and interconnected by sequence lines to indicate the order of the instruction's execution. The programming terminal receives the address of the instructions being executed and identifies this address to a displayed icon to modify the icon thus indicating the internal operation of the program on a real time basis with little interruption of the control process.

This invention is a programming aid for troubleshooting motion control programs of servo motors and has very little to do with the present invention.

US4843566

A system for controlling a mechanical manipulator in a laboratory environment employs a system of taught motions, attribute operators and procedure rules to simplify the programming task for scientists or engineers skilled in their own fields and not robotics.

The intention of this disclosure differs from the present invention in that the type and level of behaviour are different. For instance, this disclosure is concerned with moving robot manipulators and does not consider human-like behaviour such as, for example, facial expression, gestures, and so forth. Further, there is no sentence structure for programming behaviour.

US6004021

This is a toy system for use with a computer having a display. The system includes a plurality of interconnecting hardware toy pieces and a toy design system software for operating the computer. An inventory database stores an inventory of the toy pieces. An inventory database manager module updates the inventory database in response to player input. A design layout module creates and modifies a toy layout in response to player input using software representations corresponding to the inventory of the toy pieces and conforming to a plurality of design rules, and generates a screen indicating the toy layout. A layout database stores the toy layout created by the player using the toy design system. Other features include inventory management and control allowing a layout based on a fixed inventory, a layout completion module, a layout library module, a design assistance module, a simulation module and an education module. The hardware pieces may be moveable elements, such as a ball, and tracks, support pieces and stunt pieces, for guiding the ball.

This is hardware for assembling generic elements to create a toy and bears relationship to programming behaviour.

CN1212169

The electronic toy includes an external shell, a battery placed in the interior of the external shell for supplying electric power, a four-bit microcontrol unit for controlling the components of the toy, IR emitter for emitting IR signal to other toys, IR receiver for receiving IR signals from other toys, also includes four small keyboards respectively used for making commitment, canceling, selection of leftwards and rightwards movement of the cursor by means of keys, reset switches for restarting the game, LCD display for displaying these selections; and the specifications of a visitor from another planet, such as their age, weight and health, and so forth. It also has a buzzer for producing music and alarm signals. The electronic toy also contains a program for a game of culturing visitors from another planet.

US5724074

A control program for a programmable toy (24) is created using a graphic programming system implemented on a personal computer (10). In the preferred embodiment, the programmable toy comprises a vehicle that can be commanded to make turns in 45° increments and to track an infrared source (220). To create the graphic control program, the user simply selects graphic objects from a control panel (154) presented on a cartoon-like graphic screen. For example, the user can select a horn graphic object (168) or a headlight graphic object (166) to respectively activate a horn or headlights on the programmable toy. The user creates a graphic control program on a grid (160) using a mouse (23) to select the graphic objects and indicate parameters. Once the graphic control program is completed, it is downloaded from the RS-232 serial port of the personal computer into the programmable toy through a connecting link (25), which is then uncoupled from the programmable toy. A switch (29) on the toy activates the control program. Also included in the system is a calibration template (200) calibrating the programmable toy to turn accurately on a specific surface.

This has limited motion/behaviour capabilities. It does not provide a language for narrating stories or behaviour. It's only similarity to the present invention in that it provides a visual tool for programming toys.

Alice is a three-dimension, virtual-world environment with a simple graphical user interface ("GUI") for animating objects in the virtual world. The GUI provides buttons for primitive operations such as moving and rotating objects. In addition, it has "Python" (trademark), an object-oriented language for more complex animation that is not possible with the GUI alone. However, one must have a good knowledge of graphics and programming to be able to achieve decent character animation and behaviour in Alice.

Jack is a sophisticated human modeling and simulation software environment developed in the University of Pennsylvania. The sophistication of Jack does not lend itself for behaviour programming and story creation, for children and lay persons.

Summary of the invention

The present invention is intended to specify spatial and temporal unfolding of life-like behaviour in synthetic creatures, while imitating the behaviour, actions and expressions of humans, and other living creatures. It is intended to be a relatively simple and relatively easily understandable language for children, and other people who may be uninitiated in programming, to be able to exercise their creativity, and obtain some enjoyment.

The present invention may cover some or all the following behaviour categories:

- Speech and Speech Intonation – includes words, phrases and sentences that the synthetic creature utters to communicate with other humans, other creatures, or other synthetic creatures. Speech intonations are tonal variations in speech that emphasize and express the emotions of the synthetic creature.

- Facial Expression – includes various facial expressions that convey emotions such as happiness, sadness, surprise, anger, delight, fear, love and so forth.

- Gesture – includes various hand, head and other body part, gestures.

- Gaze – includes movements of the eyes in order to look at an object, human, creature or synthetic creature or look away from it, and also to communicate with it.

- Motion – includes movement from one location to another, approaching or retreating, entering or departing, crawling, walking, jogging, running, skipping, and so forth.

It is preferred that the present invention is implemented in Extensible Mark-up Language (XML).

The mark up language implementation of the present invention may include, in addition to text, graphics, audio, video and any other multimedia information.

Using the present invention, or a visual programming tool based on the present invention, a person may create a story or sequence of actions of one or more synthetic creatures in a graphical or physical environment. This may be downloaded from a network such as the Internet so that people can share stories, scripts, or the like.

The present invention may be a high-level language to script a story or set of behaviours. Preferably the present invention is implemented in the target environment, graphical or physical, using its own lower level language by a third party software or hardware company.

When using the present invention, a person may write a script to specify the behaviour of a toy or any other type of synthetic creature. The software within the toy is expected to carry out the actions as specified by the script based on its own capabilities. The present invention, or an authoring tool developed based on the present invention, may be used to script a sequence of behavior of one or more graphical cartoon characters. Such a script can then be viewed elsewhere by using a browser that understands the present invention to carry out the graphical animation of the cartoon.

Although the present invention is primarily directed at human-like behaviour, it is intended to include behaviour of other living creatures.

Description of the drawings

In order that the present invention may be readily understood and put into practical effect there shall now be described by way of non-limitative example only a preferred embodiment of the present invention, the description being with reference to the accompanying illustrative drawing, in which:

Figure 1 is a block diagram of the software architecture for implementing the present invention; and

Figure 2 is a preferred Visual Behaviour Programming Tool.

Description of preferred embodiment

The present embodiment has tags or notations for narrating a story or a small episode of behaviour of humans and other living creatures such as, for example, dogs and cats. The version described here has only human behaviour elements. However, the present invention is intended to include behaviour of other living creatures.

The present embodiment is an open, flexible, extensible language for creating and, possibly, exchanging stories for replaying in disparate mediums such as graphical animation, virtual worlds, interactive toys and robots.

A story in the present invention is denoted by the tag <story>. A story may or may not have a title. A story is defined as one or more consecutive scenes and scripts, denoted by <scene> and <script>, respectively. Each scene must have a script following it.

A scene specifies the initial arrangement of the synthetic creature or creatures, and other objects, for a script. A scene is specified before a corresponding script. Several corresponding scenes and scripts form a story. As such, a <scene> is a collection of synthetic creatures with their relative positions. Synthetic creatures or other objects are represented graphically by two dimensional or three dimensional objects, or physically by toys or robots. These synthetic objects are denoted by the tag <anyObject>. <anyObject> may represent a synthetic creature such as a man, woman, child, dog, cat, and so forth; or an inanimate object such as a table, chair, house, apple, and so forth. Every anyObject has attributes, name and objectID to uniquely identify the object. The relative position of an object with respect to another object in a scene is denoted by a <prepositionPhrase>. One or more prepositionPhrases can be used to accurately specify the relative position of an object. Preposition consists of either a <measure> or a <preposition>. Measure is the distance in certain units such as, for example, centimeters, meters, and so forth; in a certain direction such as, for example, left, right with respect another anyObject. Preposition on the other hand is an English preposition word, such as near, above, below, and so forth, to describe the location of an object relative to another object.

A script consists of a collection of sentences that describe the behaviour or actions of synthetic creatures and other objects in sequence (one after another) or together (at once). Behaviours or actions that take place in sequence or serially are enclosed within a block denoted by the <doSerial> tag. Behaviours or actions that take place simultaneously or in parallel are enclosed within a block denoted by the <doParallel> tag. A <sentence> describes a single behaviour or action of a synthetic object. For example, "Monkey ate the apple" is a valid sentence. A sentence in the present invention consists of one mandatory <subject>, one mandatory <verb>, one optional <indirectObject>, one optional <directObject> and one or more optional <prepositionPhrase>, in that order.

A subject is an anyObject. A verb denotes a behaviour or action of the subject. It should be noted that verbs of the present tense form are sufficient to describe stories and behaviour for the purpose of replaying them in graphical or physical environments. All past tense and future tense sentences can be re-framed into the present tense without loss of effect.

Following are the major classes of verbs that can be specified:

- Speech and Speech Intonation – includes verbs that denote how the speech is delivered in order that the synthetic creature communicates with other objects and/or synthetic creatures, a living creature. Speech intonations are tonal variations in speech that emphasize what is said, and express the emotions of the synthetic creature.

- Facial Expression – includes verbs that denote various facial expressions that convey emotions such as, for example, happiness, unhappiness, sadness, joy, surprise, imitation anger, delight, tenderness, fear, loathing, love, pain, detatched, enthusiastic, unpassioned, passion.

- Gestures – includes verbs that denote various hand and head gestures, and gestures of other body parts such as, for example, shoulders, arms, forearms, fingers, thumbs, chest, stomach, back, posterior, legs, ankles, feet, buttocks, eyebrows, nose, mouth.

- Gaze – includes verbs that denote movements of the eyes in order to look at an object, human creature or synthetic creature, or look away from it, and also to communicate with another synthetic creature or living creature.

- Motion – includes verbs that denote movement from one location to another including, for example, approaching, retreating, entering, arriving, leaving, departing, crawling, walking, jogging, running, skipping, swimming, flying, falling, talking, trembling, jumping, tumbling, exercising, and so forth.

Based on the above classification, there are five types of verbs:

<speechVerb>, <facialExpressionVerb>, <gestureVerb>, <gazeVerb> and <motionVerb>. Some common verbs are listed in Table 1 at the end of this description. The verbs listed are the most common body motion verbs that can be used to describe behaviour or tell a story. Gesture and gaze verbs were chosen from the book "Encyclopaedia of World Gestures" by Don Morrison. Naturally, these are by way of example only and many other verbs can be included in the present invention .

In a program of the present invention, a user can include text strings for dialogue or words spoken by the subject enclosed within the speechVerb tags that denote how the dialog is to be delivered. When the story is played or enacted in a graphical or physical environment, the words or other sounds spoken or otherwise created may be delivered using a Text-to-Speech-Synthesizer.

It is to be noted that the present invention is not limited to speech in the English language. The present invention includes speech specification and synthesis in other languages including (but not limited to) Hindi, French, German, Spanish, Portuguese, Italian, Greek, Arabic, Chinese, Japanese, and so forth. The language for any particular item of speech may be user-selected. In order to facilitate this, the present invention may include an attribute for the speechVerb called "spokenLanguage". An implementation of this in XML could be:

<!ATTLIST speechVerb spokenLanguage (english|hindi|german|french|chinese) CDATA #IMPLIED>.

There are a number of attributes common to all the verbs. All the verbs may have an adverbSpeed property, that denotes the speed with which the behaviour or action is to be carried out. There may be three degrees to the adverbSpeed: slow, normal and fast, with normal being the default value. However, other values may be used such as, for example, moderate, accelerate, and so forth. Another common attribute is the numberOfTimes to denote how many times is the action carried out. The default may be one.

IndirectObject and directObject are anyObjects that are based on the definition of indirect objects and direct objects in English grammar. According to correct English grammar, two kinds of objects follow verbs: direct objects and indirect objects. To determine if a verb has a direct object, one would isolate the verb and make it into a question by placing "whom?" or "what?" after it. The answer, if there is one, is the direct object. An indirect object is the recipient of the direct object, in a sense. To determine if a verb has an indirect object, one would isolate the verb and ask "to whom?", "to what?", "for whom?", or "for what?" after it. The answer is the indirect object. In the example below, 'gives' is the verb; 'Sheela' is the indirect object; and 'bouquet of flowers' is the direct object.

Example: Mike gives Sheela a bouquet of flowers.

Using the sentence structure described above, various types of sentences can be formed that describe actions, as illustrated below:

- Subject & Verb Sentence
  Example: Child cries.
  Here 'child' is the subject and 'cries' is the verb.

- Subject, Verb & DirectObject Sentence
  Example: Monkey eats a banana.
  'Monkey' is the subject, 'eats' is the verb, 'banana' is the DirectObject.

- Subject, Verb, DirectObject & IndirectObject Sentence
  Example: Mike gives Alice a bouquet of flowers.
  'Mike' is the subject, 'gives' is the verb, 'Alice' is the IndirectObject', and 'a bouquet of flowers' is the DirectObject.

- Subject, Verb, DirectObject, IndirectObject & PrepositionPhrase Sentence
  Example: Mike gives Alice a bouquet of flowers near the window.
  'Mike' is the subject, 'gives' is the verb, 'Alice' is the IndirectObject'. 'a bouquet of flowers' is the DirectObject, and 'near the window' is the prepositionPhrase. In the prepositionPhrase, 'near' is the preposition and 'window' is anyObject.

- Subject, Verb & PrepositionPhrase Sentence
  Example: Mike eats inside the kitchen.
  'Mike' is the subject, 'eats' is the verb, and 'inside the kitchen' is the prepositionPhrase.

It is to be noted that the present invention is not limited to the use of English language in specifying the story or behaviour. The present invention could be used to describe a story or specify behaviour in other languages of the world, such as those described earlier. The language may be user-selected. In order to facilitate this, in one embodiment of the invention, the <story> tag could have an attribute called "storyLanguage" with one language to be selected from the attribute list. For example, the XML definition for this could be as follows:

```
<!ATTLIST story storyLanguage (english|hindi|french|german|chinese) CDATA
#IMPLIED>
```

The structure of a sentence in the script then depends on the language selected. Different languages of the world may have different sentence structure and grammar. The present invention could be implemented with language structures for different languages. Although, in the present embodiment given, only English language implementation is shown, the invention is not limited to English language as such.

It is preferred that the present invention is implemented in XML. An example of the Document Type Definition (DTD) is given in Table 2.

To illustrate how the present invention can be used to program a story, Table 3 is a brief story, and Table 4 is a program of the folktale in accordance with the present invention.

Figure 1 shows the block diagram of the software architecture of the present invention.

Although a simple, high-level markup language, it may be even simpler for children and non-programmers to use a visual tool for programming. Such a visual tool can provide buttons, icons and menu items for creating a story. The visual tool can then generate the script as output. Programmers, on the other hand, can use their favourite text editor to create the program.

The program so created will then be validated and parsed by an XML parser. There are many commercially, as well as freely available XML parsers. For example, MSXML from "Microsoft" (trademark).

If the program is valid as determined by the parser, no error messages are generated, and the parser generates a tree of objects. If the program is not valid, the parser generates relevant error messages that will need to be corrected in the visual tool or the text editor.

The next step is the enactment of the story in the target environment, whether graphical or physical. For example, an Alice-Python Code Generator could be used that would take the tree of objects as input, and generate Python code for execution under the Alice graphical virtual world. As an illustration, Table 5 shows sample Python code for Alice, for a human walking routine. This routine will be called when the "walk" verb is used in a story.

Similarly, a Lego Mindstorms code generator could be used that would generate RCX code or NQC code or LegOS code for excution in the Lego Programmable Bricks.

The architecture is extensible so that future virtual worlds or toy environments can be implemented.

A visual behaviour programming tool ("VBPT") can be used to create stories visually. The VBPT can generate the code when a story is created. The code can then be fed to the XML parser for further processing as described above. Such a tool is shown in Figure 2. It has a menu bar 10; scene preparation window 12; script creation window 14; program window 16; and a story display window 18.

The menu bar 10 is used to start a story with a title; create multiple scenes and scripts; number them in order so as to sequence them; save a story; open an existing story; and so forth. The menu bar 10 also has menu items for choosing functions available as icons and buttons in other windows described below.

As the name suggests the scene preparation window 12 provides tools for preparing a scene before programming the script in the story. The window 12 has buttons and icons 20 to add synthetic objects into a scene, and to move and rotate the objects around in order to position them where necessary. The story display window 18 can display the objects in the scene as they are moved in order to provide visual feedback to the user. This is similar to the user interface in the Alice Virtual World and other visual tools.

The script preparation window 14 and program window 16 are used hand-in-hand for programming behaviour of synthetic creatures. The script preparation window 14 has four main boxes 22, 24, 26 and 28 arranged from top to bottom on the right hand side of the VBPT: program control box 22, objects box 24, verbs box 26, prepositions box 28; as well as speech text input window 30. Buttons and icons 32 are provided in the boxes (except in the speech text input 30 box) to enable a user to click on an icon or button and choose an appropriate program item. Each tool box 24, 26, 28 has arrows 34 to browse the items available, as there may be more items than the tool box 24, 26, 28 can display at once.

The topmost tool box is a program control box 22, and has rectangular buttons 36, 38, 40 respectively named doSerial 36, doParallel 38 and sentence 40. When any of these buttons 36, 38, 40 are clicked by the user, corresponding program items appear in the program window 16. For example, if user presses the doSerial button 36, a doSerial rectangle 42 appears in the program window 16 within which several sentences can be contained.

When the sentence button 40 is clicked, a sentence template 44 appears in the program window 16. A sentence consists of empty boxes 46 representing subject, verb, indirectObject, directObject, preposition and object tuples, each tuple representing a prepositionPhrase. To fill up the template, the user would have to click on one of these sentence elements 46 and click on a corresponding icon or button in the objects tool box 24, verbs tool box 26, or preposition tool box 28.

Whenever the user chooses a speech related verb from the verbs tool box 26, the user will have to key in the dialogue or words to be uttered in the speech text box 30 at the lowest right-hand corner of VBPT.

In addition, windows may be displayed to request input depending on the type of sentence element (object, verb or preposition) chosen. For example, if a verb is chosen, a window would be displayed requesting input for adverbSpeed and numberOfTimes according to the description of verbs above.

The story display window 18 window displays the story (both the scenes and script) as it is being programmed using the VBPT. In this way, the user has immediate feedback on the story being created. Immediate feedback on what is being programmed is preferable for children and other non-programmers.

As the present invention is, in one form, a general behaviour programming language, applicable to various target environments, whether graphical or physical, the story display window 18 may be varied or adapted for the target environment. For example, VBPTs for "Alice" and "Lego" "Mindstorms" may be different to that illustrated.

Authoring tools based on the present invention may be used to program behavior. Special purpose authoring tools for different purposes such as programming behaviour in toys, and for creating graphical animations, may be used.

With the present invention, the interactive toys landscape may change.

Toy makers will be able to create and manufacture toys with advanced capabilities such as mobility, expression, sensing, speech, vision, and so on. They may be able to write the software layer that can understand the instructions of the present invention and carry-out relevant actions. Creative content providers may be individuals or companies that use the present invention to create interesting scenarios including toys capable of activities such as dancing and singing; toys which can play games, plays and skits, and carry-on conversation, role playing, and so on. For example, popular stories or cartoons can be directly adopted to create puppet shows. Content providers may host the scripts on their websites and toy consumers may download them into their toys to display the behavior.

Whilst there has been described in the present invention a preferred embodiment of the present invention, it will be understood by those skilled in the technology that many variation or modifications in details of design or construction may be made without departing from the present invention.

Type of Verb	Sample set of verbs
speechVerb	Say, recite, sing, scream, shout, mutter, and stutter.
motionVerb	Walk, sit, stand, run, lie down, skip, dance, throw, catch, count, look, eat, drink, drive, ride, swim, read, think, hold, lift, pull, push, drop, touch, caress, beat, pat, strike, move, roll, turn, wipe, write, pickup, putdown, place, climbup, climbdown, jump, shake, scratch, rub, stab, shrug, bite, and cover.
GestureVerb & gazeVerb	ArmsRaise, armsGrasp, armsAkimbo, armsFold, bellyPat, bellyRub, cheekKiss, cheekSlap, chestPoint, chestBeat, earCup, earThumb, earBlock, earCover, eyeWink, eyeWipe, eyeBrowFlash, eyesClose, eyesBlink, eyesStare, eyesRaise, eyesWeep, faceCover, fingersClasp, fingersUnclasp, fingerOpen, fingersClose, fingersPoint, fingersWave, fistBeat, fistClench, foreFingerBeckon, foreFingerBlow, foreFingerPoint, foreFingerRaise, foreFingerWag, foreFingerBang, hairClasp, handBeckon, handSalute, handShake, handWag, handWave, hatRaise, headNod, headShake, palmHighSlap, palmSlap, palmContact, thumbDown, thumbUp, thumbSuck, and waistBow.
facialExpression Verb	ShowNeutral, showHappiness, showSadness, showSurprise, showAnger, showFear, showDespair, and showDisappointment.

TABLE 1

TABLE 2

```
<?xml version = "1.0" standalone="yes"?>
<!DOCTYPE story [
<!ELEMENT story (scene, script)+>
<!ATTLIST story title CDATA #IMPLIED>
<!ELEMENT scene ((anyObject|defObject), prepositionPhrase*)*>
<!ELEMENT anyObject EMPTY>
<!ATTLIST anyObject objectId ID #REQUIRED>
<!ATTLIST anyObject objName CDATA #REQUIRED>
<!ELEMENT prepositionPhrase (location, (anyObject|defObject))>
<!ELEMENT defObject EMPTY>
<!ATTLIST defObject ref IDREF #REQUIRED>
<!ELEMENT location (measure|preposition)>
<!ELEMENT measure EMPTY>
<!ATTLIST measure direction (left|right|top|bottom|front|back) #REQUIRED>
<!ATTLIST measure integerDistance CDATA "1">
<!ATTLIST measure unit (CM|MT|KM) "KM" >
<!ELEMENT preposition EMPTY>
<!ATTLIST preposition word
(near |from |above |across |between |on |under |through |next_to |of |at
|behind |about |around |over |besides |into |by |beyond |below |inside)
#REQUIRED>
<!ELEMENT script (doSerial | doParallel)+>
<!ELEMENT doSerial (sentence)+>
<!ELEMENT doParallel (sentence)+>
<!ELEMENT sentence (subject, verb, indirectObject?,
directObject?, prepositionPhrase*)>
<!ELEMENT subject (anyObject|defObject)>
<!ELEMENT verb (speechVerb|motionVerb| gestureGazeVerb|
facialExpressionVerb)>
<!ATTLIST verb adverbSpeed (slow|normal|fast) "normal">
<!ATTLIST verb numberOfTimes CDATA "0">
<!ELEMENT speechVerb (#PCDATA)>
<!ATTLIST speechVerb verb
(say|recite|sing|scream|shout|mutter|stutter|exclaim) "say">
<!ELEMENT motionVerb EMPTY>
<!ATTLIST motionVerb motion
(walk| sit| stand| run| lie_down| skip| dance| throw| catch| count| look|
eat| drink| drive| ride| swim| read| think| hold| lift| pull| push| drop|
touch| caress| beat| pat| strike| move| roll| turn| wipe| write| pickup|
putdown| place| climbup| climbdown| jump| shake| scratch| rub| stab|
shrug| bite| cover|open|give|take) #REQUIRED>
<!ELEMENT gestureGazeVerb EMPTY>
<!ATTLIST gestureGazeVerb gestureGaze (ArmsRaise| armsGrasp| armsAkimbo|
armsFold| bellyPat| bellyRub| cheekKiss| cheekSlap| chestPoint| chestBeat|
earCup| earThumb| earBlock| earCover| eyeWink| eyeWipe| eyeBrowFlash|
eyesClose| eyesBlink| eyesStare| eyesRaise| eyesWeep| faceCover|
fingersClasp| fingersUnclasp| fingerOpen| fingersClose| fingersPoint|
fingersWave| fistBeat| fistClench| foreFingerBeckon| foreFingerBlow|
foreFingerPoint| foreFingerRaise| foreFingerWag| foreFingerBang| hairClasp|
handBeckon| handSalute| handShake| handWag| handWave| hatRaise|
headNod| headShake| palmHighSlap| palmSlap| palmContact| thumbDown|
thumbUp| thumbSuck| waistBow) #REQUIRED>
<!ELEMENT facialExpressionVerb EMPTY>
<!ATTLIST facialExpressionVerb facialExpression
(showNeutral|showHappiness|showSadness|showSurprise
|showFear|showAnger|showDespair) "showNeutral">
<!ELEMENT indirectObject (anyObject|defObject)>
<!ELEMENT directObject (anyObject|defObject)>
]>
```

TABLE 2 (CONCLUSION)

The Purse of Gold

A beggar found a leather purse that someone had dropped in the marketplace. Opening it, he discovered that it contained 100 pieces of gold. Then he heard a merchant shout, "A reward! A reward to the one who finds my leather purse!"

Being an honest man, the beggar came forward and handed the purse to the merchant saying, "Here is your purse. May I have the reward now?"

"Reward?" scoffed the merchant, greedily counting his gold. "Why the purse I dropped had 200 pieces of gold in it. You've already stolen more than the reward! Go away or I'll tell the police."

"I'm an honest man," said the beggar defiantly. "Let us take this matter to the court."

In court the judge patiently listened to both sides of the story and said, "I believe you both. Justice is possible! Merchant, you stated that the purse you lost contained 200 pieces of gold. Well, that's a considerable cost. But, the purse this beggar found had only 100 pieces of gold. Therefore, it couldn't be the one you lost."

And, with that, the judge gave the purse and all the gold to the beggar.

TABLE 3

TABLE 4

```
<story title="The Purse Of Gold">
  <scene>
    <anyObject objectId="o1" objName="sky" />
    <anyObject objectId="o2" objName="beggar" />
      <prepositionPhrase>
        <location>
          <measure direction="bottom" integerDistance="10" />
        </location>
        <defObject ref="o1" />
      </prepositionPhrase>
    <anyObject objectId="o3" objName="merchant" />
      <prepositionPhrase>
        <location>
          <measure direction="left" integerDistance="2" />
        </location>
        <defObject ref="o2" />
        </prepositionPhrase>

  </scene>
  <script>
    <doSerial>
      <sentence>
        <subject >
          <anyObject  objectId="o4" objName="purse" />
        </subject>

        <verb>
            <motionVerb motion="drop" />
        </verb>
        <prepositionPhrase>
            <location>
              <preposition word="on" />
            </location>
                <anyObject objectId="o5" objName="ground" />
        </prepositionPhrase>
        <prepositionPhrase>
          <location>
            <preposition word="from" />
          </location>
            <anyObject objectId="o6" objName="pocket" />
        </prepositionPhrase>

    </sentence>
    <sentence>
      <subject >
        <defObject ref="o2" />
    </subject>

    <verb>
        <motionVerb motion="walk" />
    </verb>
  <prepositionPhrase>
    <location>
    <measure direction="front" integerDistance="6" unit="CM" />
    </location>
    <defObject ref="o4" />
  </prepositionPhrase>

</sentence>

<sentence>
  <subject >
    <defObject ref="o2" />
  </subject>
  <verb>
    <motionVerb motion="look" />
  </verb>
  <directObject>
    <defObject ref="o4" />
  </directObject>
</sentence>

<sentence>
  <subject >
    <defObject ref="o2" />
  </subject>
  <verb>
    <motionVerb  motion="pickup" />
  </verb>
  <directObject >
    <defObject ref="o4" />
  </directObject>
</sentence>

<sentence>
  <subject >
    <defObject ref="o2" />
  </subject>
  <verb>
    <motionVerb  motion="open" />
  </verb>
  <directObject>
    <defObject ref="o4" />
  </directObject>
</sentence>

<sentence>
  <subject >
    <defObject ref="o2" />
  </subject>
  <verb>
```

TABLE 4 (PART)

```
          <speechVerb  verb="exclaim">
              "This purse contains 100 pieces of gold !"
          </speechVerb>
      </verb>

    </sentence>

    <sentence>
      <subject >
          <defObject ref="o3" />
      </subject>
      <verb>
          <speechVerb  verb="shout">
                "A reward! A reward to the one who finds my leather
                purse!"
          </speechVerb>
        </verb>
    </sentence>

    <sentence>
      <subject >
        <defObject ref="o2" />
      </subject>
      <verb>
        <motionVerb motion="walk" />
      </verb>
      <prepositionPhrase>
        <location>
          <preposition word="near" />
        </location>
        <defObject ref="o3" />
      </prepositionPhrase>
    </sentence>

</doSerial>

<doParallel>
    <sentence>
      <subject >
        <defObject ref="o2" />
      </subject>
      <verb>
        <motionVerb motion="give" />
      </verb>
      <indirectObject >
        <defObject ref="o3" />
      </indirectObject>
      <directObject >
        <defObject ref="o4" />
      </directObject>

    </sentence>
    <sentence>
```

TABLE 4 (PART)

```
         <subject >
            <defObject ref="o2" />
         </subject>
         <verb>
            <speechVerb verb="say">
               "Here is your purse. May I have the reward now?"
            </speechVerb>
         </verb>
      </sentence>
      <sentence>
         <subject >
            <defObject ref="o3" />
         </subject>
         <verb>
            <motionVerb motion="take" />
         </verb>
         <directObject >
            <defObject ref="o4" />
         </directObject>

      </sentence>

</doParallel>
<doSerial>
      <sentence>
      <subject >
         <defObject ref="o3" />
      </subject>
      <verb>
         <motionVerb motion="open" />

      </verb>
      <directObject >
         <defObject ref="o4" />
      </directObject>
      </sentence>
      <sentence>
      <subject >
         <defObject ref="o3" />
      </subject>
      <verb>
         <speechVerb verb="scream" >
            " Reward ? Why the purse I dropped had 200 pieces of gold in it.
            You've already stolen more than the reward! Go away or I'll tell the
            police."
         </speechVerb>
      </verb>

      </sentence>
      <sentence>
      <subject >
         <defObject ref="o2" />
```

TABLE 4 (PART)

```
    </subject>
    <verb>
      <speechVerb verb="say" >
      " I'm an honest man, Let us take this matter to the court."
      </speechVerb>
    </verb>

    </sentence>

</doSerial>
</script>

<scene >
  <defObject ref="o3" />
  <prepositionPhrase>
    <location>
    <preposition word="on" />
    </location>
    <defObject ref="o5" />
  </prepositionPhrase>
  <defObject ref="o3" />
  <prepositionPhrase>
    <location>
      <measure direction="left" />

    </location>
  <defObject ref="o2" />
  </prepositionPhrase>
  <anyObject objectId="o7" objName="judge" />
  <prepositionPhrase>
      <location>
          <measure direction="front" />

      </location>
  <defObject ref="o2" />
  </prepositionPhrase>

</scene>
<script>
  <doParallel>
    <sentence>
      <subject >
        <defObject ref="o2" />
      </subject>
      <verb>
        <motionVerb motion="look" />

      </verb>
      <directObject >
        <defObject ref="o7" />
      </directObject>
```

TABLE 4 (PART)

```
    </sentence>
    <sentence>
    <subject >
      <defObject ref="o2" />
    </subject>
    <verb>
      <speechVerb verb="say" >
      "Sir , I found this purse on the road in the market place .It had 100
      pieces of gold . I heard this merchant shout that he would give reward
      to anyone who finds and gives back his purse. I honestly returned his
      purse . But he didn't give my reward ".
      </speechVerb>
    </verb>
    </sentence>

</doParallel>
  <doParallel>
    <sentence>
      <subject >
        <defObject ref="o3" />
      </subject>
      <verb>
        <motionVerb motion="look" />

      </verb>
        <directObject >
          <defObject ref="o7" />
        </directObject>

    </sentence>

    <sentence>
      <subject >
        <defObject ref="o3" />
      </subject>
      <verb>
        <speechVerb verb="say" >
          "The purse contained two-hundred pieces of gold. The beggar stole
          100 pieces of gold. So the beggar got his reward and there is now
          no need to reward him"
        </speechVerb>
      </verb>
    </sentence>
</doParallel>

<doSerial>
    <sentence>
      <subject >
         <defObject ref="o7" />
      </subject>
      <verb>
```

TABLE 4 (PART)

```
                <speechVerb verb="say">
                    "I believe you both. Justice is possible! Merchant, you stated that
                    the purse you lost contained 200 pieces of gold. Well, that's a
                    considerable cost. But, the purse this beggar found had only 100
                    pieces of gold. Therefore, it couldn't be the one you lost."
                </speechVerb>
             </verb>
        </sentence>
        <sentence>
            <subject >
                <defObject ref="o7" />
            </subject>
            <verb>
                 <motionVerb motion="give" />
            </verb>
            <indirectObject >
                 <defObject ref="o2" />
            </indirectObject>
            <directObject >
                  <defObject ref="o4" />
            </directObject>

        </sentence>
</doSerial>
<doParallel>
        <sentence>
            <subject >
                    <defObject ref="o2" />
            </subject>
            <verb>
                <facialExpressionVerb facialExpression="showHappiness" />

            </verb>
        </sentence>
        <sentence>
            <subject >
                    <defObject ref="o3" />
            </subject>
            <verb>
                    <facialExpressionVerb facialExpression="showSadness" />

            </verb>
        </sentence>

      </doParallel>

    </script>
</story>
```

TABLE 4 (CONCLUSION)

TABLE 5

```
#
#Walking Routine
#

def Rstep(person):
        anim = DoInOrder (
        DoTogether (
          #move leftarm forward
          person.larm.turn(backward, 1/8, Duration=0.1),

          #rleg step forward
          person.dress.rthigh.turn(backward, 1/8, Duration=0.1),
          person.dress.rthigh.rleg.turn(forward, 1/16, Duration=0.1)
        ),

        #person move and lleg lift
        DoTogether (
          person.dress.lthigh.lleg.turn(forward, 1/8, Duration=0.1),
          person.move(forward, 1/4, Duration=0.1)
          ),

        #rleg come back to position
        person.dress.rthigh.turn(forward, 1/8, Duration=0.1),
        person.dress.rthigh.rleg.turn(backward, 1/16, Duration=0.1),

        DoTogether (
          #lleg come back to position
          person.dress.lthigh.lleg.turn(backward, 1/8, Duration=0.1),

          #move leftarm back to position
          person.larm.turn(forward, 1/8, Duration=0.1)
        ),

        )
        return anim.Start()

def Lstep(person):
        anim = DoInOrder (

        DoTogether (
          #move rightarm forward
          person.rarm.turn(backward, 1/16, Duration=0.1),

          #lleg step forward
          person.dress.lthigh.turn(backward, 1/8, Duration=0.1),
          person.dress.lthigh.lleg.turn(forward, 1/16, Duration=0.1)
        ),

        #move and rleg lift
        DoTogether (
          person.dress.rthigh.rleg.turn(forward, 1/8, Duration=0.1),
          person.move(forward, 1/4, Duration=0.1)
          ),
            #lleg come back to position
            person.dress.lthigh.turn(forward, 1/8, Duration=0.1),
            person.dress.lthigh.lleg.turn(backward, 1/16, Duration=0.1),

            DoTogether (
              #rleg come back to position
              person.dress.rthigh.rleg.turn(backward, 1/8, Duration=0.1),

              #move rightarm back to position
              person.rarm.turn(forward, 1/16, Duration=0.1)
            ),

    )
            return anim.Start()

    def Walking(person):
            anim = DoInOrder (
              Rstep(person),
              Lstep(person))
            return anim.Start()

    def Walk(person, steps):
            anim = loop(Walking(person), steps)
            return anim.Start()
```

TABLE 5 (CONCLUSION)

THE CLAIMS:

1. A method for programming at least one synthetic creature (as defined herein) using a markup language to enable the synthetic creature to imitate the behaviour actions and expressions of humans and other living creatures, the method including the steps of:
   (a) creating a story, the story including at least one scene and at least one script;
   (b) the at least one scene including the at least one synthetic creature;
   (c) the script including at least one of a series of sentences;
   (d) each sentence including at least one subject and at least one action the subject is to perform.

2. A method as claimed in claim 1, wherein the at least one action is determined by a verb selected from the group including:
speech and speech intonation, facial expression, gestures, gaze, and motion.

3. A method as claimed in claim 2, wherein speech and speech intonation include verbs that denote how the speech is delivered in order that the synthetic creature communicates with at least one other object.

4. A method as claimed in claim 3, wherein the other object is another synthetic creature.

5. A method as claimed in any one of claims 2 to 4, wherein speech intonation includes tonal variations in speech to give emphasis to at least one aspect of the speech and to express at least one emotion of the synthetic creature.

6. A method as claimed in any one of claims 2 to 5, wherein facial expression includes verbs that denote various facial expression used to convey emotions.

7. A method as claimed in claim 6, wherein the emotion is selected from the

list including happiness, unhappiness, sadness, joy, surprise, anger, delight, tenderness, fear, loathing, love, pain, detached, enthusiastic, unfeeling, imitation, impassioned, and passion.

8. A method as claimed in any one of claims 2 to 7, wherein gestures include verbs that denote gestures of body parts of the synthetic creature.

9. A method as claimed in claim 8, wherein the body parts are selected from the list including head, shoulder, arms, forearms, hands, fingers, thumbs, chest, stomach, back, posterior, legs, ankles, feet, buttocks, eyebrows, nose, mouth.

10. A method as claimed in any one of claims 2 to 9, wherein gaze includes verbs that denote movements of the eyes in order to look at another object and to thereby communicate with the other object.

11. A method as claimed in any one of claims 2 to 10, wherein motion includes verbs that denote movement from one location to another location.

12. A method as claimed in claim 11, wherein the motion is selected from the list including approaching, retreating, entering, arriving, leaving, departing, arriving, crawling, walking, jogging, running, skipping, swimming, flying, falling, talking, trembling, tumbling, exercising, jumping, and rotating.

13. A method as claimed in any one of claims 2 to 12, wherein all verbs have an adverb speed property that denotes the speed with which the action is to be performed.

14. A method as claimed in claim 13, wherein the speed is selected from the list including slow, normal, fast, moderate and accelerate.

15. A method as claimed in any one of claims 1 to 14, wherein the sentence includes at least one direct object.

16. A method as claimed in any one of claims 1 to 15, wherein each sentence

includes at least one indirect object.

17. A method as claimed in any one of claims 1 to 16, wherein each sentence includes at least one preposition phrase.

18. A method as claimed in any one of claims 1 to 17, wherein the at least one scene also includes other objects with their relative positions.

19. A method as claimed in any one of claims 1 to 18, wherein the program is implemented in XML.

20. A method as claimed in any one of claims 1 to 19, wherein the story is written in a user-selected language.

21. A method as claimed in any one of claims 1 to 19, wherein after completion the program is parsed by a parser.

22. A method as claimed in any one of claims 3 to 5, wherein the speech may be in a user-selected language.

23. A graphic user interface to enable the method of any one of claims 1 to 22 to be performed, the graphic user interface including:

(a) a story display area;
(b) a script preparation area;
(c) a scene preparation area; and
(d) a speech text area.

24. A graphic user interface as claimed in claim 23, wherein the story display area enables the objects in a scene to be displayed to provide visual feedback to a user.

25. A graphic user interface as claimed in claim 23 or claim 24, wherein the scene preparation area has a plurality of tools to enable an object to be included in the scene, for the object to be located in a required position, and for the object to be oriented in a required manner.

26. A graphic user interface as claimed in claim 25, wherein the plurality of

tools include buttons and icons.

27. A graphic user interface as claimed in any one of claims 23 to 26, wherein the script preparation area has a plurality of sub-areas for programming behaviour of a least one synthetic creature.

28. A graphic user interface as claimed in claim 27, wherein the plurality of sub-areas include a program control box, an objects box, a verbs box, and a prepositions box.

29. A graphic user interface as claimed in claim 27 or claim 28, wherein each of the plurality of sub-areas includes buttons and icons to enable a user to select an appropriate program item.

30. A graphic user interface as claimed in claim 28 or claim 29, wherein each of the plurality of sub-areas has arrows to enable a user to browse available items.

31. A graphic user interface as claimed in claim 28, wherein the program control box includes first and second areas to enable actions to be performed serially and in parallel respectively.

32. A graphic user interface as claimed in claim 28 or claim 31, wherein the program control box includes a third area for creating sentences.

33. A graphic user interface as claimed in claim 32, wherein when the third area is activated by a user, a template is displayed for the creation of a sentence.

34. A graphic user interface as claimed in claim 33, wherein the template includes regions for each of: subject, verb, indirect object, direct object, preposition and object tuples.

35. A graphic user interface as claimed in claim 34, wherein each preposition and object tuples represents a preposition phrase.

36. A graphic user interface as claimed in any one of claims 27 to 35,

wherein upon any one or more of the plurality of sub-areas being activated it causes an appropriate display in a program area.

37. A computer markup language for the programming of a synthetic creature (as defined herein), the language including:

(a) at least one object which is to perform at least one action, the object and the action arranged as a sentence;

(b) a plurality of sentences being arranged as a script;

(c) a scene corresponding to a script and including the at least one synthetic creature;

(d) the scenes and scripts together creating a story.

38. A language as claimed in claim 37, wherein the at least one object includes the synthetic creature.

39. A language as claimed in claim 37 or claim 38, wherein the at least one action in defined by a corresponding verb.

40. A language as claim in claim 39, wherein the at least one a verb is selected from the group including: speech and speech intonation, facial expression, gestures, gaze, and motion.

41. A language as claimed in claim 40, wherein speech and speech intonation include verbs that denote how the speech is delivered in order that the synthetic creature communicates with at least one other object.

42. A language as claimed in claim 41, wherein the other object is another synthetic creature.

43. A language as claimed in any one of claims 40 to 42, wherein speech intonation includes tonal variations in speech to give emphasis to at least on aspect of the speech and to express at least one emotion of the synthetic creature.

44. A language as claimed in any one of claims 40 to 43, wherein facial expression includes verbs that denote various facial expression used to convey emotions.

45. A language as claim in claim 44, wherein the emotion is selected from the list including happiness, unhappiness, sadness, joy, surprise, anger, delight, tenderness, fear, loathing, love, pain, detached, enthusiastic, unfeeling, imitation, impassioned, and passion.

46. A language as claimed in any one of claims 40 to 45, wherein gestures include verbs that denote gestures of body parts of the synthetic creature.

47. A language as claimed in claim 46, wherein the body parts are selected from the list including head, shoulder, arms, forearms, hands, fingers, thumbs, chest, stomach, back, posterior, legs, ankles, feet, buttocks, eyebrows, nose, mouth.

48. A language as claimed in any one of claims 40 to 47, wherein gaze includes verbs that denote movements of the eyes in order to look at another object and to thereby communicate with the other object.

49. A language as claimed in any one of claims 40 to 48, wherein motion includes verbs that denote movement from one location to another location.

50. A language as claimed in claim 49, wherein the motion is selected from the list including approaching, retreating, entering, arriving, leaving, departing, crawling, walking, jogging, running, skipping, swimming, flying, falling, talking, trembling, tumbling, exercising, jumping, and rotating.

51. A language as claimed in any one of claims 40 to 51, wherein all verbs have an adverb speed property that denotes the speed with which the action is to be performed.

52. A language as claimed in claim 51, wherein the speed is selected from the list including slow, normal, fast, moderate and accelerate.

53. A language as claimed in any one of claims 37 to 52, wherein the sentence includes at least one direct object.

54. A language as claimed in any one of claims 37 to 53, wherein the sentence includes at least one indirect object.

55. A language as claimed in any one of claims 37 to 54, wherein the sentence includes at least one preposition phrase.

56. A language as claimed in any one of claims 37 to 55, wherein the language is implemented in XML.

57. A language as claimed in any one of claims 37 to 56, wherein the scene includes other objects and their relative positions.

58. A language as claimed in any one of claims 37 to 57, wherein the story is written in a user-selected language.

59. A language as claimed in any one of claim 41 to 43, wherein the speech may be a user-selected language.

60. A language as claimed in any one of claims 35 to 55, wherein the language includes one or more selected from the list including text, images, audio, video, and other media formats.

61. A programmable storage device readable by a machine, including a program of executable instructions to perform the method of any one or more of claims 1 to 22.

62. A programmable storage device readable by a machine, including a program of executable instructions to display on a display screen a graphical user interface in accordance with any one or more of claims 23 to 36.

63. A programmable storage device readable by a machine, including a program of executable instructions in the language of any one or more claims 37 to 60.

64. A graphical user interface in accordance with any one of claims 23 to 36

when used to perform the method of any one of claims 1 to 22.

65. A method as claimed in any one of claims 1 to 22 using the language of any one of claims 37 to 60.

66. A language as claimed in any one of claims 36 to 60 to perform the method of any one of claims 1 to 22 on a graphical user interface of any one of claims 23 to 36.

67. A programmable storage device readable by a machine, including a program of executable instructions in the language of any one or more of claims 37 to 60 to perform the method of any one or more claims 1 to 22 on a graphical user interface of any one or more of claims 23 to 36.

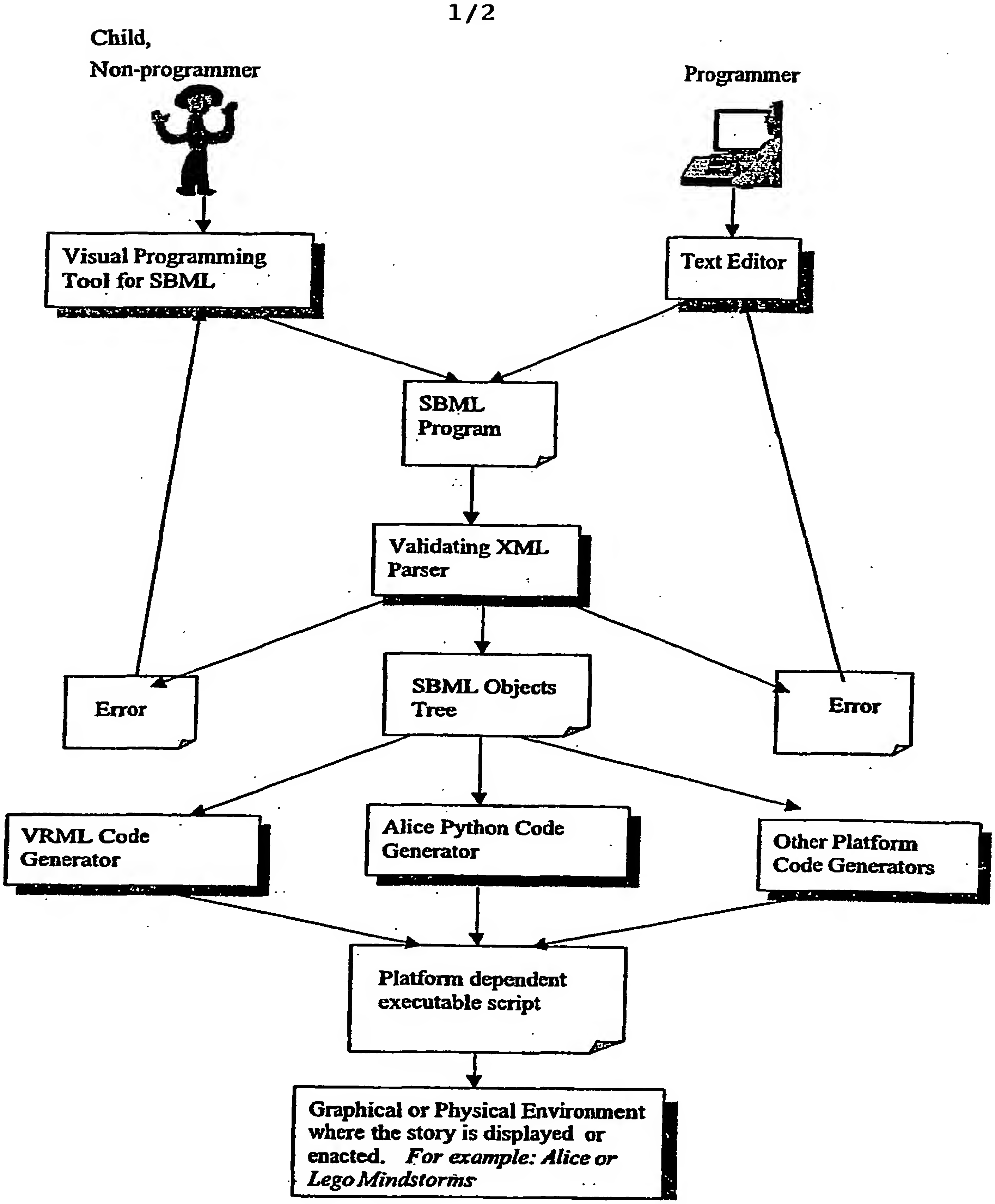
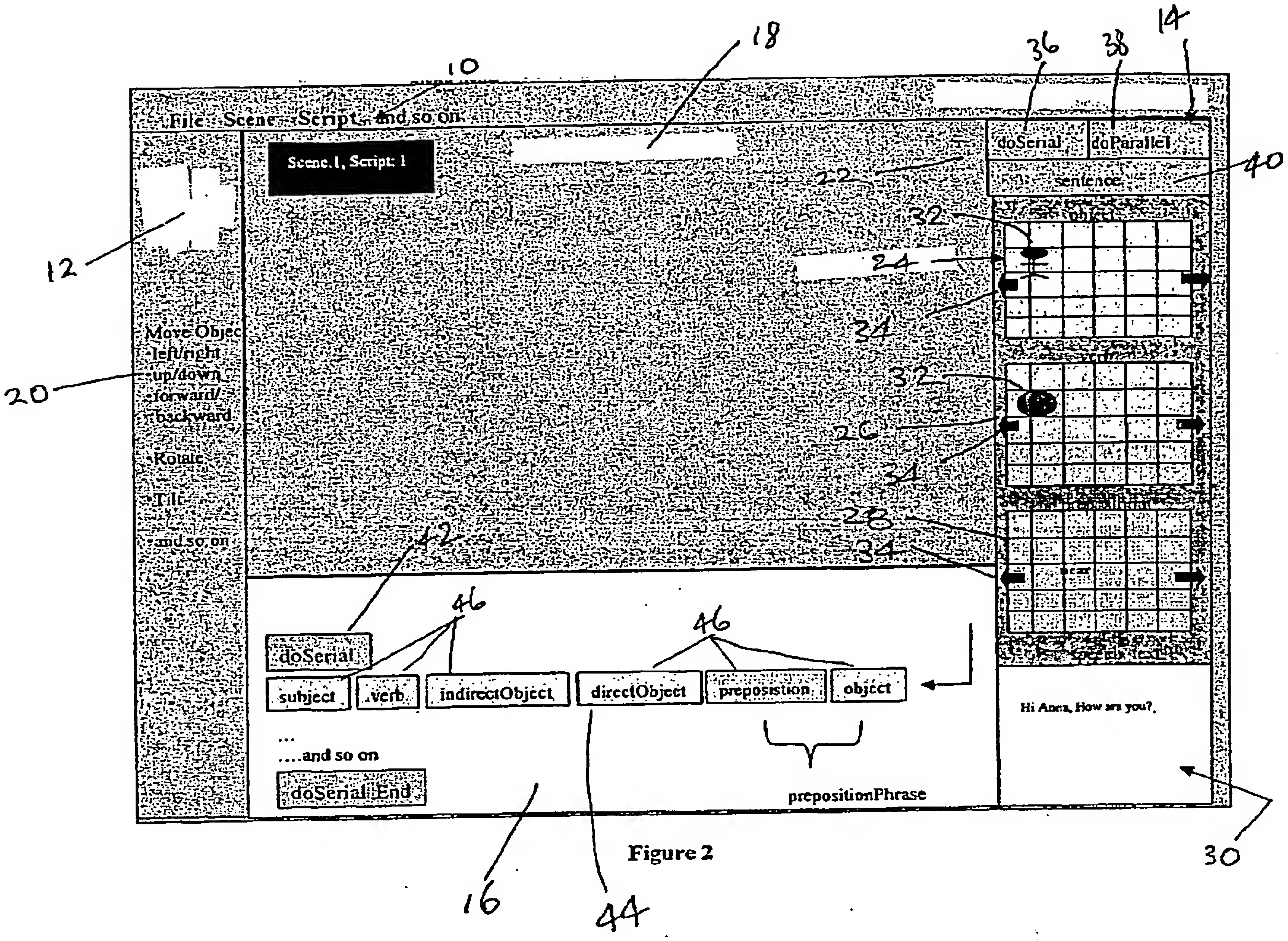

Child,
Non-programmer
Programmer
Visual Programming Tool for SBML
Text Editor
SBML Program
Validating XML Parser
Error
SBML Objects Tree
Error
VRML Code Generator
Alice Python Code Generator
Other Platform Code Generators
Platform dependent executable script
Graphical or Physical Environment where the story is displayed or enacted. *For example: Alice or Lego Mindstorms*

Figure 1



Hi Anna, How are you?

Figure 2

INTERNATIONAL SEARCH REPORT

International Application No
PCT/S 0/00166

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06N3/00 G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06N G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, WPI Data, PAJ, IBM-TDB, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	MAKATCHEV M ET AL: "Human-robot interface using agents communicating in an XML-based markup language" PROCEEDINGS 9TH IEEE INTERNATIONAL WORKSHOP ON ROBOT AND HUMAN INTERACTIVE COMMUNICATION. IEEE RO-MAN 2000, OSAKA, JAPAN, 27-29 SEPTEMBER 2000, pages 270-275, XP002171714 2000, Piscataway, NJ, USA, IEEE, USA ISBN: 0-7803-6273-X	1,19,37, 56
A	the whole document	2-18, 20-36, 38-55, 57-67

	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 11 July 2001	Date of mailing of the international search report 24/07/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Schenkels, P

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 98 06043 A (UNIV NEW YORK ;PERLIN KENNETH (US); GOLDBERG ATHOMAS (US)) 12 February 1998 (1998-02-12)	1,19,37, 56
A	abstract	2-18, 20-36, 38-55, 57-67
	page 3, line 31 -page 31, line 29; figures 1-7 ---	
A	SIMMONS R ET AL: "A task description language for robot control" PROCEEDINGS. 1998 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS. INNOVATIONS IN THEORY, PRACTICE AND APPLICATIONS ,VICTORIA, BC, CANADA, 13-17 OCT 1998, pages 1931-1937 vol.3, XP002171715 1998, New York, NY, USA, IEEE, USA ISBN: 0-7803-4465-0 the whole document ---	1,37
A	EP 0 992 927 A (KONAMI CO LTD) 12 April 2000 (2000-04-12) column 1, line 3 -column 23, line 51; figures 1-11 ---	1,37
A	FUJITA M ET AL: "AN OPEN ARCHITECTURE FOR ROBOT ENTERTAINMENT" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS,US,NEW YORK, ACM, vol. CONF. 1, 5 February 1997 (1997-02-05), pages 435-442, XP000775167 ISBN: 0-89791-877-0 the whole document -----	1,37

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/S[illegible]0/00166

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9806043	A	12-02-1998	EP 0919031	A	02-06-1999
EP 0992927	A	12-04-2000	JP 2000107442	A	18-04-2000